



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/001,586	10/31/2001	David J.C. Johnson	10013442-1	1948

7590 11/12/2004
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER

ELMORE, REBA I

ART UNIT PAPER NUMBER

2187

DATE MAILED: 11/12/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/001,586

Applicant(s)

JOHNSON ET AL.

Examiner

Reba I. Elmore

Art Unit

2187

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 10 June 2004.
2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-4 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) 1-4 is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-4 are presented for examination.

Specification

2. The specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Claims 1-4 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 6-13 of copending Application No. 10/004193, claims 1-8 of Application No. 10/004195 and claims 1-10 of Application No. 10/001594. Although the conflicting claims are not identical, they are

not patentably distinct from each other because of the reasoning given below in the comparison of the compared claims. A detailed comparison has been written for the claims in the present invention in conflict with the claims of the conflicting applications based on obviousness type double patenting with all of the claims of the conflicting application.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

10/001586

1. A method of evicting an entry in a cache memory, comprising:

setting a bit to a first state when the entry is accessed;

setting the bit to a second logical state; and

evicting the entry when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

2. A method of evicting an entry in a cache memory, comprising:

setting a bit to a first logical state when the entry is written;

setting the bit to a second logical state; and

10/004193

6. A method of limiting dirty entries in a cache memory, comprising:

incrementing a counter when an entry in the cache memory transitions to a modified state;

decrementing the counter when a modified entry is evicted from the cache memory;
setting a bit to a first state and then a second state is equivalent to either incrementing or decrementing a counter dependent upon the type of access to a cache memory entry

evicting at least one modified entry from the cache memory when a count in the counter exceeds a predetermined threshold.
with the predetermined threshold being equivalent to the predetermined time

6. A method of limiting dirty entries in a cache memory, comprising:

incrementing a counter when an entry in the cache memory transitions to a modified state;

decrementing the counter when a modified entry is evicted from the cache memory;

evicting the entry when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

3. A method of evicting at least one entry in a set of entries in a cache memory, comprising:

setting a bit to a first logical state when the entry corresponding to an index is accessed;

setting the bit to a second logical state; and

evicting the entry when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

4. A method of evicting at least one entry in a set of entries in a cache memory, comprising:

setting a bit to a first logical state when the entry corresponding to an index is accessed;

setting the bit to a second logical state; and

setting a bit to a first state and then a second state is equivalent to either incrementing or decrementing a counter dependent upon the access to a cache memory entry being a write

evicting at least one modified entry from the cache memory when a count in the counter exceeds a predetermined threshold.
with the predetermined threshold being equivalent to the predetermined time

6. A method of limiting dirty entries in a cache memory, comprising:

incrementing a counter when an entry in the cache memory transitions to a modified state;

decrementing the counter when a modified entry is evicted from the cache memory;
setting a bit to a first state when an index is accessed with accessing an index being equivalent to the comparison which is necessarily performed when a cache is accessed to determine whether or not an entry is in the cache and setting the bit to a second logical state being equivalent to either incrementing or decrementing a counter dependent upon the type of access to a cache memory entry

evicting at least one modified entry from the cache memory when a count in the counter exceeds a predetermined threshold.
with the predetermined threshold being equivalent to the predetermined time

6. A method of limiting dirty entries in a cache memory, comprising:

incrementing a counter when an entry in the cache memory transitions to a modified state;

decrementing the counter when a modified entry is evicted from the cache memory;

setting a bit to a first state when an index is accessed with accessing an index being equivalent to the comparison which is necessarily performed when a cache is accessed to determine whether or not an entry is in the cache and setting the bit to a second logical state being equivalent to either incrementing or decrementing a counter dependent upon the type of access to a cache memory entry

evicting the entry when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

evicting at least one modified entry from the cache memory when a count in the counter exceeds a predetermined threshold.
with the predetermined threshold being equivalent to the predetermined time

10/001586

1. A method of evicting an entry in a cache memory, comprising:

setting a bit to a first state when the entry is accessed;

setting the bit to a second logical state; and

evicting the entry when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

2. A method of evicting an entry in a cache memory, comprising:

setting a bit to a first logical state when the entry is written;

10/004195

1. A method of improving performance of a computer system, comprising:

a cache memory system inherently sets bits when entries in the cache are accessed with the setting of the bits being either a first or second logical state as either a one or zero

(b) evicting, from a cache memory, at least one entry that has remained unchanged for at least the time period;

(a) specifying a time period;

(c) measuring at least one performance parameter;

(d) changing the value of the time period;

(e) repeating steps (b) and (c); and

(d) determining whether the performance parameter has changed.

1. A method of improving performance of a computer system, comprising:

a cache memory system inherently sets bits when entries are written in the cache with the setting of bits being either a first or

setting the bit to a second logical state; and

evicting the entry when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

3. A method of evicting at least one entry in a set of entries in a cache memory, comprising:

setting a bit to a first logical state when an entry corresponding to an index is accessed;

setting the bit to a second logical state; and

evicting at least one entry corresponding to the index when the bit is at the second logical state after at least predetermined time after being set to the second logical state.

4. A method of evicting at least one entry in a set of entries in a cache memory, comprising:

setting a bit to a first logical state when an entry corresponding to an index is modified;

setting the bit to a second logical state; and

second logical state as either a one or zero

(b) evicting, from a cache memory, at least one entry that has remained unchanged for at least the time period;

(a) specifying a time period;

(c) measuring at least one performance parameter;

(d) changing the value of the time period;

(e) repeating steps (b) and (c); and

(d) determining whether the performance parameter has changed.

1. A method of improving performance of a computer system, comprising:

a cache memory system inherently set bits when entries are written in the cache with the setting of bits being either a first or second logical state as either a one or zero and the accessing of an index being equivalent to the comparison which is necessarily performed when a cache is accessed to determine whether or not an entry is in the cache

(b) evicting, from a cache memory, at least one entry that has remained unchanged for at least the time period;

(a) specifying a time period;

(c) measuring at least one performance parameter;

(d) changing the value of the time period;

(e) repeating steps (b) and (c); and

(d) determining whether the performance parameter has changed.

1. A method of improving performance of a computer system, comprising:

a cache memory system inherently set bits when entries are modified in the cache with the setting of bits being either a first or second logical state as either a one or zero

and the accessing of an index being equivalent to the comparison which is necessarily performed when a cache is accessed to determine whether or not an entry is in the cache

evicting at least one entry corresponding to the index when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

(b) evicting, from a cache memory, at least one entry that has remained unchanged for at least the time period;

(a) specifying a time period;

(c) measuring at least one performance parameter;

(d) changing the value of the time period;

(e) repeating steps (b) and (c); and

(d) determining whether the performance parameter has changed.

10/001586

1. A method of evicting an entry in a cache memory, comprising:

setting a bit to a first state when the entry is accessed;

setting the bit to a second logical state; and

evicting the entry when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

2. A method of evicting an entry in a cache memory, comprising:

10/001594

7. A method of detecting whether an entry in a cache memory has been recently accessed, comprising:

setting a bit to a first logical state when the entry is accessed;

setting the bit to a second logical state; and

determining that the entry has not been recently accessed when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

evicting the entry is obvious over the determining of the entry having not been recently accessed as using the LRU or least recently used algorithm for cache eviction methods as the LRU replacement algorithm is common in the cache arts and frequently used for that purpose

8. A method of detecting whether an entry in a cache memory is dirty and has not been

setting a bit to a first logical state when the entry is written;

setting the bit to a second logical state; and

evicting the entry when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

3. A method of evicting at least one entry in a set of entries in a cache memory, comprising:

setting a bit to a first logical state when an entry corresponding to an index is accessed;

setting the bit to a second logical state; and

evicting at least one entry corresponding to the index when the bit is at the second logical state after at least predetermined time after being set to the second logical state.

4. A method of evicting at least one entry in a set of entries in a cache memory, comprising:

recently accessed, comprising:

setting a bit to a first logical state when the entry is written;

setting the bit to a second logical state; and

determining that the entry is dirty and has not been recently accessed when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

evicting the entry is obvious over the determining of the entry having not been recently accessed as using the LRU or least recently used algorithm for cache eviction methods as the LRU replacement algorithm is common in the cache arts and frequently used for that purpose

9. A method of detecting whether at least one entry in a set of entries in a cache memory has been recently accessed, comprising:

setting a bit to a first logical state when an entry corresponding to an index is accessed;

setting the bit to a second logical state; and

determining that at least one entry corresponding to the index has not been recently accessed when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.

evicting the entry is obvious over the determining of the entry having not been recently accessed as using the LRU or least recently used algorithm for cache eviction methods as the LRU replacement algorithm is common in the cache arts and frequently used for that purpose

10. A method of detecting whether at least one entry in a set of entries in a cache

	memory is dirty and has not been recently accessed, comprising:
setting a bit to a first logical state when an entry corresponding to an index is modified;	setting a bit to a first logical state when an entry corresponding to an index is modified;
setting the bit to a second logical state; and	setting the bit to a second logical state; and
evicting at least one entry corresponding to the index when the bit is at the second logical state after at least a predetermined time after being set to the second logical state.	determining that at least one entry corresponding to the index is dirty and has not been recently accessed when the bit is at the second logical state after at least a predetermined time after being set to the second logical state. <i>evicting the entry is obvious over the determining of the entry having not been recently accessed as using the LRU or least recently used algorithm for cache eviction methods as the LRU replacement algorithm is common in the cache arts and frequently used for that purpose</i>

5. The present claimed invention and the conflicting claims are not exactly the same, however, the sets of claims are not patentably distinct. These differences are not sufficient to render the claims patentable and distinct and therefore a terminal disclaimer is required (*Georgia Pacific Corp v United States Gypsum Co.*, 52 USPQ2d 1590, US Court of Appeals Federal Circuit 1999).

6. “A latter patent claim is not patentable distinct from an earlier patent claim if the latter claim is obvious over, or anticipated by, the earlier claim. In *re Longi*, 759 F.2d at 896, 225 USPQ at 651 (affirming a holding of obvious-type double patenting because the claim at issue were obvious over claims in four prior art patents); In *re Berg*, 140 F.3d at 1437, 46 USPQ2d at 1233 (Fed. Cir. 1998) (affirming a holding of obvious-type double patenting where a patent application claim to a genus is anticipated by a patent claim to a species within that genus). *ELI LILLY AND COMPANY v BARR LABORATORIES*,

INC., United States Court of Appeals for the Federal Circuit, ON PETITION FOR REHEARING EN BANC (DECIDED: May 30, 2001).

35 USC § 102

7. The rejection of claims 1-4 as being anticipated by Dean et al. is ***maintained*** and repeated below.

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

9. Claims 1-4 are rejected under 35 U.S.C. 102(e) as being anticipated by Dean et al. (US 2002/0152361 A1)

10. Dean teaches the invention (claim 1) as claimed including a method of evicting an entry in a cache memory comprising:

setting a bit to a first logical state when the entry is accessed as setting the age bits to NEW (e.g., see paragraph 0038);

setting the bit to a second logical state as setting the age bits to a minimum value (e.g., see paragraph 0038); and,

evicting the entry when the bit is at the second logical state after at least a predetermined time after being set to the second logical state as using the aging of the bits to determine the next line to evict (e.g., see paragraph 0038) with the predetermined time relating to a time stamp or age value (e.g., see paragraphs 0024-0025).

11. Dean teaches the invention (claim 2) as claimed including a method of evicting an entry in a cache memory comprising:

setting a bit to a first logical state when the entry is written as setting the age bits to NEW (e.g., see paragraph 0038);

setting the bit to a second logical state as setting the age bits to a minimum value (e.g., see paragraph 0038); and,

evicting the entry when the bit is at the second logical state after at least a predetermined time after being set to the second logical state as using the aging of the bits to determined the next line to evict (e.g., see paragraph 0038) with the predetermined time relating to a time stamp or age value (e.g., see paragraphs 0024-0025).

12. Dean teaches the invention (claim 3) as claimed a method of evicting at least one entry in a set of entries in a cache memory comprising:

setting a bit to a first logical state when an entry corresponding to an index which is part of the cache address (e.g., see Figure 1) is accessed as setting the age bits to NEW (e.g., see paragraph 0038);

setting the bit to a second logical state as setting the age bits to a minimum value (e.g., see paragraph 0038); and,

evicting at least one entry corresponding to the index when the bit is at the second logical state after at least a predetermined time after being set to the second logical state as using the aging of the bits to determined the next line to evict (e.g., see paragraph 0038) with the predetermined time relating to a time stamp or age value (e.g., see paragraphs 0024-0025).

13. Dean teaches the invention (claim 4) as claimed including a method of evicting at least one entry in a set of entries in a cache memory comprising:

setting a bit to a first logical state when an entry corresponding to an index which is part of the cache address (e.g., see Figure 1) is modified as setting the age bits to NEW (e.g., see paragraph 0038);

setting the bit to a second logical state as setting the age bits to a minimum value (e.g., see paragraph 0038); and,

evicting at least one entry corresponding to the index when the bit is at the second logical state after at least a predetermined time after being set to the second logical state as using the aging of the bits to determine the next line to evict (e.g., see paragraph 0038) with the predetermined time relating to a time stamp or age value (e.g., see paragraphs 0024-0025).

Response to Applicant's Remarks

14. Applicant's arguments filed June 10, 2004 have been fully considered but they are not persuasive.

15. As to the reference not teaching a bit within the field for an entry being evicted when a logical state has not changed for a predetermined time, this limitation is taught to the extent required by the actual claim language. The replacement policy includes using a line age field which is reset when the line has been stored in the cache for a predetermined duration of time and has become the next line to be evicted when a replacement line is being entered in the cache.

Art Unit: 2187

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

17. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Reba I. Elmore, whose telephone number is (571) 272-4192. The examiner can normally be reached on M-TH from 7:30am to 6:00pm, EST.

If attempts to reach the examiner by telephone are unsuccessful, the art unit supervisor for AU 2187, Donald Sparks, can be reached for general questions concerning this application at (571) 272-4201. Additionally, the official fax phone number for the art unit is (703) 746-7239.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Tech Center central telephone number is (571) 272-2100.



Reba I. Elmore
Primary Patent Examiner
Art Unit 2187

November 4, 2004